THE
PLOTTER

CLACKAMAS COMPUTER APPLIED TRAINING SOCIETY NEWS LETTER

****

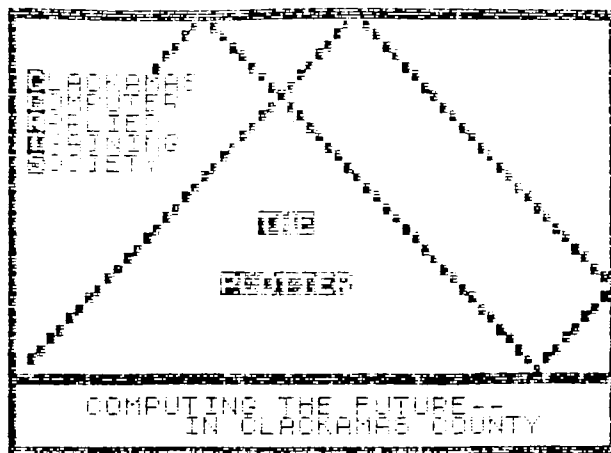VOLUME 12   **   NUMBER 5

******

MAY   1994

COMPUTING THE FUTURE--
IN CLACKAMAS COUNTY

MEETING

The MAY meeting will be:

on: SUN., MAY 15, 1994

MEETING   2:00 TO 5:00 P.M.
at: Rod Gowen's home
     14784 S. Quail Circle
   Oregon City

WHAT YOU WILL FIND IN THIS ISSUE:

FROM THE EDITOR'S DESK

This month we provide an article on basic BASIC programming that hopefully will interest some of our readers. The author's name, Jim Wagner, will be recognized as the provider of a write-up on his book, The Amateur Packet Radio Handbook in a recent newsletter.

A little background on Jim. He is my eldest son of 50+ years, has a doctorate in electronics plus degrees in physics and electrical engineering. While being employed some 30 years ago at Tektronics in product development, he took an early course in computer programming that I well remember. The instruction was with teletype machines using punch tape! The computer was someplace in Texas, being accessed with a code name via the telephone system. Charges were based on computer time so there was real incentive to not waste time with poor programming.

This article was written as words of advice from son to father in an effort to get me to do better programming. He is well aware of my struggles with the T/S 1000, 2068, PC clone with GW

»»»

CLACKAMAS
COMPUTER
APPLIED
TRAINING
SOCIETY
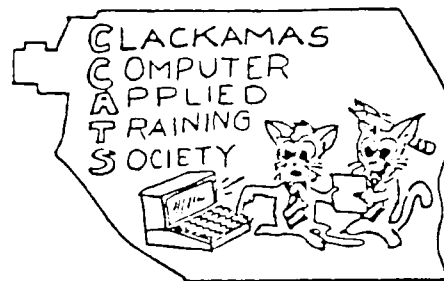
BASIC. I must say he has a low opinion of my early computers plus reluctance to learn machine language programming. Accordingly I have deleted some personal references from the script.

We have a program in QBASIC as promised last month. This is my first effort at converting GW BASIC to QBASIC. There will probably be some glaringly poor programming but at least it works. I hope to find a good guide in this language.

Rod Gowan makes reference to doing his income taxes, Federal and State, with the PC program CA-SIMPLY TAX. I also used this program with satisfactory results. The IRS-approved printed forms looked a little odd in comparison witht those provided by IRS. I had already obtained another tax program but it took too much memory for my 268 computer. I did use the Oregon state program from this company, taking Fed. data from the CA program. I had also calculated my tax on paper and found I had shorted myself $58 in data that showed up on the computer program. The Computer Associates Fed. program was priced right, postage and handling.

Has anyone come up with a tray for accurately using a hand scanner, other than the commercial producr that sell for $40? I would like to make one if there is a good alternative. Maybe we could run a construction article.

Our soon-to-be-released publication is now ready for a second test printing after making a few corrections, mainly in the page numbering. After assembling a book it was apparent that these corrections were required. We are all pleased with the end product of the efforts of half of our membership.

PRINTER INK CLEANER

Dick Wagner

Those who have re-inked larger printer ribbons know that it is almost impossible to use re-inking equipment without getting ink on one or more fingers. Rod Gowen, RMG Enterprises, recommends using WD 40 lubricant as a hand cleaner.

Recently I had re-inked a 17+ yd ribbon for my Epson printer but neglected to clean up after the operation. So seeing this "uncleaned" equipment on my work bench I proceded to drain the plastic ink holder and in the process really blackened several fingers. The WD 40 was out of reach but I had a tube of hand cleaner near by used to clean up after working on the car. Being the experimental type I squirted a small amounf of GO-JO hand cleaner on the fingers and spread it around as it liquified. Wiping with a paper towel left my hands CLEAN. Further washing with detergent and water left my hands "dainty clean" and spotles!

This hand cleaner can usually be purchased at auto parts stores. Ask for GO-JO Industrial Stock #0906. According to the label of contents, which I finally got around to reading, it consists of Isoparaffins, Water, Mineral Oil, Soaps, Surfacants, Propylene Glycol, Aloe Extract, Biotin, Colors, Fragarance, Lanolin, Niacinamide, Petrolium, Preservatives, Vitamine E, & Wheat Germ Extract. Just compare that with WD 40 and see why it is so good. Priced about $1.00 for 5 fluid oz..

The directions state that GO-JO thoroughly and safely cleans (removes?): adhesive, asphalt, carbon, cement, graphite, grease, grime, mastic, paint, printer's ink, putty, shellac, stain, tar, & many more.

OBSCURE AGES

Dick Wagner

The puzzle this month is best solved by using linear equations in which variables are multiplied by constants and added together. This is the method used in the program. As noted, the original program is from the book, BRAIN GAMES FOR KIDS AND ADULTS, written for Apple computers.

The program selects 3 ages by using the integers of RND calculations. This selection assures that the ages progress in the proper order. The player inputs the 3 ages in order as requested. Keep in mind that Ron is oldest and Pat is youngest.

We hope the reader works this out as we just may later have one or 2 of a more difficult nature. Maybe this is one for the math student in the family to try before exposing the program.

```
10 REM source, BRAIN GAMES FOR
ADULTS AND KIDS, for Apple comp
uter
20 REM converted by Dick Wagne
r, 4/94
30 REM to Sinclair BASIC
40 REM program uses linear equ
ations to arrive at solution.
100 LET P=INT (10*RND*(1)+5)
110 LET J=P+INT (20*RND*(1))
120 LET R=J+INT (20*RND*(1))
125 REM these 3 equations will
always assure that J is larger t
han P and R is always larger tha
n J
130 PRINT "RON IS ";R-J;" YEARS
OLDER THAN JIM": PRINT
140 PRINT "RON IS ";R-P;" YEARS
OLDER THAN PAT": PRINT
150 PRINT "THEIR COMBINED AGES
IS ";P+J+R: PRINT
155 PRINT "WHAT ARE THEIR AGES?
": PRINT
160 INPUT "ENTER PAT,S AGE ";A:
PRINT "PAT'S AGE IS ";A
170 INPUT "ENTER JIM,S AGE ";B:
PRINT "JIM'S AGE IS ";B
180 INPUT "ENTER RON'S AGE ";C:
PRINT "RON'S AGE IS ";C
190 IF P=A AND J=B AND R=C THEN
GO TO 220
200 PRINT "WRONG, IT IS ";P;","
;J;",";R
210 GO TO 230
220 PRINT "RIGHT ON, GOOD"
230 FOR T=1 TO 1000: NEXT T: GO
TO 240
240 PRINT : PRINT "AGAIN? PRESS
Y OR N";: INPUT A$
250 IF A$="" THEN GO TO 250
255 PRINT : PRINT : CLS
260 IF A$="Y" OR A$="y" THEN GO
TO 100
```
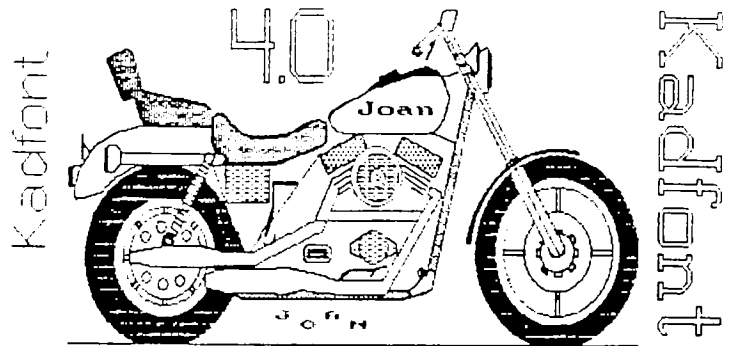

2068 PLOTTING

John McMichael has a program and hardware that interfaces a 2068 computer and a Commodore 1520 plotter. The two lower images are a plot of a Kawaski motor cycle, the smaller image a 1X scale and the lower image 2X scale. Note that the image is slightly large for the paper width.

Oddly, the 2X image was first generated by drawing short vertical lines, then the horizontal lines were added. Note that the image is almost a reversal of the small image as lined seem to outline the solid colors.
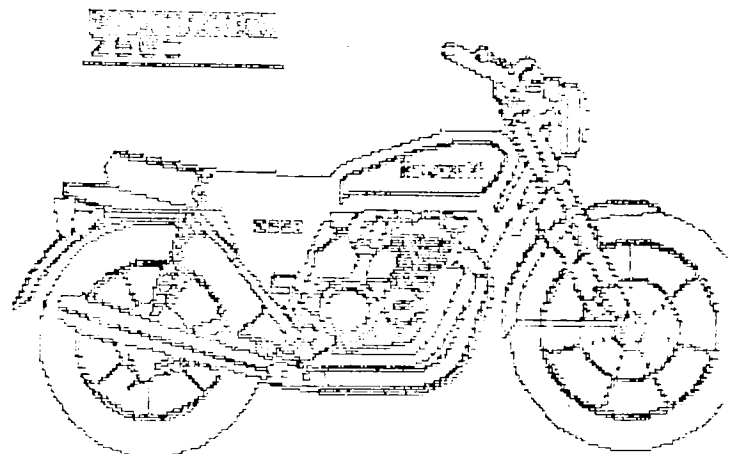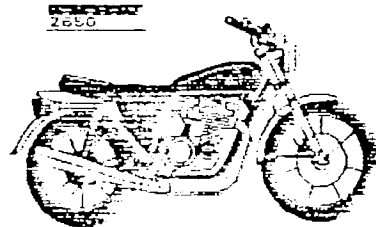
RMG UPDATE NEWS FOR MAY 1994
VOLUME 6, NUMBER 5

** RMG NEWS **

TAX TIME IS HERE! By the time you read this, April 15th will havepassed and, hopefully, most of us are done with our 1993 income taxes! How many of you did them on a computer this year? What better use could there be for a computer? Isn't that what they are good at--crunchingnumbers? I have been doing my taxes on the PC for about 7-8 years now and have been sticking with the same program for most of that time. It started with a company whose name you may remember--TIMEWORKS. The began creating software for the T/S computers and went to the Commodore and then on to the IBM clones. The program first appeared for the Commodore 64 called SWIFTAX. It then changed to EASYTAX. In 1992 it was sold to SOFTKEY SOFTWARE who kept the EASYTAX name. In 1993, SOFTKEY sold it to COMPUTER ASSOCIATES who now call it CA-SIMPLY TAX. For the first time since the Commodore, there is now a version for Oregon State taxes available. Upon running the software, 6 bugs were found. Three were found by others and CA notified us. The other 3 caused me to spend about 3-5 hours on the phone with the IRS and CA tech support to get my taxes done. Hopefully,by next year, there will be fewer bugs. We can only wait and see. Due to the fact that this year was more complicated than previous years due to moving, buying a home, etc., it would have taken me longer than usual anyway. In years past it never seemed to take me more than a couple of hours to do my return, once all of the data was at hand. AND, the program prints out IRS acceptable forms, even on a 9 pin printer! Watch for a review of this software elsewhere.

Summer is almost here! How's the yard looking? I can't believe that we are still seeing snow and ice in the midwest and plains. Even in the high passes in Oregon last week. Let us hope that it is over for the summer!

Our work on the book, THE BEST OF THE PLOTTER, is almost completed! We should be approving the final proofs by the time you are reading this. It has been a long time coming and a lot of work for a few people. We hope that it will be worth the wait to you, the buyers! If you are still interested in this book, please let us know.

Are you a member of T/SNUG yet? If not, why not? I just received and read the latest issue of ZXir Clive Alive, the newsletter and it was full of interesting information, programs, articles and ads from many TS suppliers. For the $10 membership fee, this has to be the best bargain going for us TS users! If you need an address to join, just let me know.

KEEP WATCHIN' FOR MORE NEWS! Rod Gowen, wner,RMGEnterprises
14784 South Quail Grove Circle, Oregon City, OR 97045
503/655-7484 8AM-6PM PT * FAX/VOICEMAIL: 503/655-4116 24 HRS

Much of what I list below is intended to improve programming "reliability". You cannot have a program which works well and is free of bugs if you cannot understand it later and if you cannot do it close to right the first time. Thus, many of these things are designed to put up roadblocks to errors. They also make programs larger. But larger is not necessarily bad.

It is particularly important to recognize that these points are well recognized by professional programmers. There is no reason for amateurs to avoid them. In the end, the only difference between software written by an amatuer and a professional should be whether or not the writer was paid! There is no justification, what-so-ever, for amateur software to be sloppy, prone to bugs, to work poorly, etc. Sure, it is one thing if you want to write a quick program to test something or get an answer to a problem and then discard it (the program, that is). But, if you are going to spend more than an hour writing a program, it is probably in your best interests to do it better. Once you learn how, it will go faster, it will be more likely to work correctly the first time, and may actually be more fun.

A. ORGANIZATION
1. My favorite program orgainization has 3 sections
   a. Initialization, where variables are intialized, the screen is set up, etc.
   b. An EXECUTIVE whose task is to repeatedly call various subroutines.
      It just keeps looping around and around until some end condition occurs.
   c. Shutdown, where files are saved, screen is cleared, etc.
2. Some general rules I follow:
   a. Almost never stop program execution to wait for a keyboard input, serial input, etc.
   b. Be aware of possible errors, and prevent them from disrupting program execution. Examples are invalid key entry, files which cannot be found, etc.
   c. Use LOTS of white space in your program. It is much, much, easier to read and thus easier to understand when you come back to it later.
   d. Document - enough is never enough!
   e. Avoid spaghetti code and work for layered code. If it the organization looks like a layer cake, you are going in the right direction. Spaghetti code jumps hither and yon. You can never really tell where something starts or ends.
   f. Organize the program into well defined tasks which have a starting point and an ending point. Suppose that you wish to draw a "box" on the screen (perhaps to emphesize a keyboard input). Draw the whole box in one place. Don't do the top edge in one place, sides in another, and bottom in a third. Not only does this make it more difficut to understand, but it also makes it very difficult to use the same code somewhere else in the program when you need another box.
   g. There are two common programming styles, "bottom-up" and "top-down". Neither is good in isolation (though both have been heavily promoted in the past as such). Bottom-up concentrates on the foundation. You construct the tools you expect to need in order to make the program work. For example, if you expect the program to need vector sums, you would develop a subroutine to do this and test is very thoroughly. Once done, you can forget about its internal details and use it as a foundation block for higher layers of the program. Top-down focuses on the look and feel of the program. You define carefully how the program is to operate in all of the situations you can imagine. Then, you break the operation down and try to identify things which share common characteristics. For example, pop-up menus are all really the same in a program; they only differ in where they appear on the screen and what items they list. In actuality, well-written programs combine these two strategies.

B. SOME INITIAL HINTS

There are two things which happen over and over in programs. One is categorizing a value. The other is setting a variable to one of two values depending on some condition. Here are some hints for simplifying these two tasks. These hints are based on breaking the preconception about doing them which is based on "intuition" or trying to mold the computer program after the way we tend to think.

1. Conditional variable values

The intuitive solution is usually based on the following logic: Under one set of conditions, I want X to have one value and under another set of conditions, I want X to have a different value. We try to pattern the flow of the program exactly the way the statement is framed. In a BASIC which has only "IF (true-or-false statement) THEN..." and no "IF (...) THEN ... ELSE ...", we tend to write:

```
bbb IF (true-false statement) THEN GOTO eee
ccc X=onevalue
ddd GOTO nnn
eee X=other value
nnn next operation
```

This can be shortened to:

```
bbb IF (true-false statement) THEN X=othervalue; GOTO eee
ccc X=onevalue
nnn next operation
```

Our mind often tells us that if one statement is true, the program should do one thing and if the statement is not true, then the program should do another. But if we think of other implementations, an alternative is to ALWAYS set the variable to one value, then change it under a specific condition. This violates our sense of "what is right" however, because some of the time, the program does something, then immediately undoes it. This is where intuition fails. Execution speed is so fast now that its (usually) no big deal and it saves code space (by not having to jump around the other choice) to do it this way.

```
bbb X=onevalue
ccc IF (true-false statement) THEN X=othervalue
nnn next operation
```

An alternative is possible if you have the "IF (...) THEN ... ELSE ..." capability. The whole thing can be very neatly written as:

```
ccc IF (...) THEN X=onevalue ELSE X=othervalue
```

Notice, also, that the preceeding two examples DO NOT USE ANY GOTOs!

2. Categorizing a variable's value often takes a lot of code. Its not too difficult if the variable can only take on a very limited range of integer values; in this case, a "computed GOTO" can be used. But if the variable can take on "real" values (as opposed to purely integer) or if there are lots of categories, it is not so easy.

Lets take an example. Suppose that we want to see which of many "bins" X falls into. Lets name the bins as BIN1, BIN2, BIN3, ... And lets define the bins by saying that if X is between X1 and X2, it is in BIN1, if X is between X2 and X3, it is in BIN2, etc, with the assumption that X1<X2<X3<.... From the way this problem is usually stated, the way the code is usually written as:

```
aaa IF ((X>X1) AND (X<X2)) THEN BIN=BIN1; GOTO xxx
bbb IF ((X>X2) AND (X<X3)) THEN BIN=BIN2; GOTO xxx
ccc etc
```

The "problem" with this code is that every boundary is tested twice and if
the bin happens to be near the end of the list, there can be a LOT of extra
testing & time taken. But we need only take a look at what really happens to
make major simplifications. Note that line aaa removes all of Xs in bin1 so
there is no real reason to repeat the check in bbb; those were removed earlier.
A much more compact (and faster) way of writing this same task would be

```
aaa IF (X<X1)  GOTO xxx
bbb IF (X<X2)  THEN BIN=BIN1; GOTO xxx
ccc IF (X<X3)  THEN BIN=BIN2; GOTO xxx
```

What we have done is recognize the "sieving" or "filtering" nature of the process.
We know that a certain group has been eliminated by an earlier line, so there
is no reason to test all over again.

Note that all of these examples have ignored (for simplicity) what happens if X=X1, etc.
Depending on how the bins need to be defined, the "<" symbols might be replaced by
"<=". In actuality, the first example categorised X in BIN1 if it is strictly greater than
X1 and strictly less than X2 while the second categorized X in BIN1 if it is equal to
or greater than X1 but less than X2. Changing the "<" to "<=" in the second
example changes BIN1 to be all of the X's which are greater than X1 and less than or
equal to X2. Thus, this alters which boundary is contained within the bin.

CONTINUED NEXT ISSUE

```
'Dick Wagner 3/27/94
'the answer in QBASIC to last issue cockroach story
PRINT "******************************"
PRINT "*ANSWER TO THE COCKROACH STORY*"
PRINT "******************************"
PRINT
A = 25
B = A - 1 / 5 * A
C = B - 1 / 4 * B
D = C - 1 / 3 * C
E = D - 1 / 2 * D
PRINT "25 COCKROACHES MARCHING ACROSS THE FLOOR"
PRINT "YOU KILL "; A - B; " WITH FRYING PAN"
PRINT "YOU KILL "; B - C; " WITH DISHES"
PRINT "YOU KILL "; D - E; " WITH KITCHEN TABLE"
PRINT "YOU KILL "; E; " WITH KITCHEN LAMP"
PRINT "THE CAT EATS "; A - 1 / 5 * A - 1 / 4 * B - 1 / 3 * C - 1 / 2 * D;
PRINT " COCKROACHES"
PRINT "          THE END"
```

TIMEX & SINCLAIR SOFTWARE
Dick Wagner

The March issue of NUT & VOLTS magazine carries this classified ad under Computer Software:

> Commodore 64, T199/4, Timex, Sinclair, VIC 20 software programs. Send SASE for listing to: Computer Bin, Box 1826 Perris, CA 91572. IBM software send $1.00 for catalog disk.

Who knows it just might turn up some new programs so why not give it a try.

—NOTICE—

Opinions expressed in articles are not necessarily those of members of the Clackamas Computer Applied Training Society . Meeting minutes carry the consensus of members present at meeting. This newsletter nor staff will not be held liable for any damage or consequences due to following instructions, or review of products as contained in this newsletter

SUBSCRIPTION RATES

FREE SUBSCRIPTION WITH PAID
CCATS MEMBERSHIP-$20.00/YR

PLACE YOUR AD HERE

AD RATES
****
1 FOUR LINE CLASSIFIED FREE
WITH MEMBERSHIP

$ .50 / LINE
$ 5.00 / 1/4 PAGE
$ 8.00 / 1/2 PAGE
$ 15.00 / FULL PAGE

ALL AD COPY MUST BE IN BY THE 10TH OF THE MONTH PRIOR TO THE MONTH AD IS TO RUN.

CCATS
14784 S. Quail Grove Circle
Oregon City, OR 97045

CHICAGO AREA TIMEX USERS GROUP
CATUG--C/O BOB SWOGER
613 PARKSIDE CIRCLE
STREAMWOOD, IL 60107

60107-1647